LEARNING WITH LITTLE DATA: EVALUATION OF DEEP LEARNING ALGORITHMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep learning has become a widely used tool in many computational and classification problems. Nevertheless obtaining and labeling data, which is needed for strong results, is often expensive or even not possible. In this paper three different algorithmic approaches to deal with limited access to data are evaluated and compared to each other. We show the drawbacks and benefits of each method. One successful approach, especially in one- or few-shot learning tasks, is the use of external data during the classification task. Another successful approach, which achieves state of the art results in semi-supervised learning (SSL) benchmarks, is consistency regularization. Especially virtual adversarial training (VAT) has shown strong results and will be investigated in this paper. The aim of consistency regularization is to force the network not to change the output, when the input or the network itself is perturbed. Generative adversarial networks (GANs) have also shown strong empirical results. In many approaches the GAN architecture is used in order to create additional data and therefor to increase the generalization capability of the classification network. Furthermore we consider the use of unlabeled data for further performance improvement. The use of unlabeled data is investigated both for GANs and VAT.

1 INTRODUCTION

Deep neural networks have shown great performance in a variety of tasks, like speech or image recognition. However often extremely large datasets are necessary for achieving this. In real world applications collecting data is often very expensive in terms of cost or time. Furthermore collected data is often unbalanced or even incorrect labeled. Hence performance achieved in academic papers is hard to match.

Recently different approaches tackled these problems and tried to achieve good performance, when otherwise fully supervised baselines failed to do so. One approach to learn from very few examples, the so called few-shot learning task, consists of giving a collection of inputs and their corresponding similarities instead of input-label pairs. This approach was thoroughly investigated in Koch et al. (2015), Vinyals et al. (2016), Snell et al. (2017) and gave impressive results tested on the Omniglot dataset (Lake et al. (2011)). In essence a task specific similarity measure is learned, that embeds the inputs before comparison.

Furthermore semi-supervised learning (SSL) achieved strong results in image classification tasks. In SSL a labeled set of input-target pairs $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}_L$ and additionally an unlabeled set of inputs $\boldsymbol{x} \in \mathcal{D}_{UL}$ is given. Generally spoken the use of $\mathcal{D}_{UL}$ shall provide additional information about the structure of the data. Generative models can be used to create additional labeled or unlabeled samples and leverage information from these samples (Salimans et al. (2016), Odena (2016)). Furthermore in Dai et al. (2017) it is argued, that GAN-based semi-supervised frameworks perform best, when the generated images are of poor quality. Using these badly generated images a classifier with better generalization capability is obtained. On the other side Kingma et al. (2014) uses generative models in order to learn feature representations, instead of generating additional data.

Another approach in order to deal with limited data is consistency regularization. The main point of consistency regularization is, that the output of the network shall not change, when the input or the network itself is perturbed. These perturbations may also result in inputs, which are not realistic anymore. This way a smooth manifold is found on which the data lies. Different approaches to consistency regularization can be found in Miyato et al. (2018), Sajjadi et al. (2016), Laine & Aila (2017), and Tarvainen & Valpola (2017).

The aim of this paper is to investigate how different approaches behave compared to each other. Therefore a specific image and sound recognition task is created with varying amount of labeled data. Beyond that it is further explored how different amounts of unlabeled data support the tasks, whilst also varying the size of labeled data. The possible accuracy improvement by labeled and unlabeled examples is compared to each other. Since there is a correlation between category mismatch of unlabeled data and labeled data (Oliver et al. (2018)) reported, we investigate how this correlation behaves for different approaches and datasets.

2 Algorithmic Approaches

When dealing with little data, transfer learning (Yosinski et al. (2014), Bengio (2011)) offers for many use cases a good method. Transfer learning relies on transferring knowledge from a base model, which was trained on a similar problem, to another problem. The weights from the base model, which was trained on a seperate big dataset, are then used as initializing parameters for the target model. The weights of the target model are afterwards fine-tuned. Whilst often yielding good results, nevertheless a similar dataset for the training of the base model is necessary. Many problems are too specific and similar datasets are not available. In Miyato et al. (2018) transfer learning achieves better results than any compared consistency regularization method, when transferring from ImageNet (Deng et al. (2009)) to CIFAR-10 (Krizhevsky (2009)). On contrast, no convincing results could be achieved when transferring from ImageNet to SVHN (Netzer et al. (2011)), although the task itself remains a computer vision problem. Therefore the generalization of this approach is somehow limited. In order to increase the generalization of this work transfer learning is not investigated.

Instead this paper focuses on generative models, consistency regularization, and the usage of external data during the classification of new samples. Since there exist several algorithms for each of these approaches, only one representative algorithm for each of the three approaches is picked and compared against each other.

2.1 Usage of external Data during Classification

The usage of external data after training during the classification task is a common technique used in few shot learning problems. Instead of input-label pairs, the network is trained with a collection of inputs and their similarities.

Due to its simplicity and good performance the approach by Koch et al. (2015), which is inspired by Bromley et al. (2014), is used in this paper. Koch et al. (2015) uses a convolutional siamese neural network, which basically learns an embedding of the inputs. The same convolutional part of the network is used for two inputs $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. After the convolution each input is flattened into a vector. Afterwards the L_1 distance between the two embeddings is computed and fed into a fully-connected layer, which outputs a similarity between $[0, 1]$.

In order to classify a test image $\boldsymbol{x}$ into one of K categories, a support set $\{\boldsymbol{x}_k\}_{k=1}^{K}$ with examples for each category is used. The input $\boldsymbol{x}$ is compared to each element in the support set and the category corresponding to the maximum similarity is returned. When there are more examples per class the query can be repeated several times, such that the network returns the class with the highest average similarity.

Using this approach is advantageous, when the number of categories is high or not known at all. On the downside the prediction of the category depends on a support set and furthermore the computational effort of predicting a category increases with $\mathcal{O}(K)$, since a comparison has to be made for each category.

2.2 Consistency Regularization

Consistency regularization relies on increasing the robustness of a network against tiny perturbations of the input or the network. For perturbations of the input $d\left(f(\boldsymbol{x};\boldsymbol{\theta}), f(\hat{\boldsymbol{x}};\boldsymbol{\theta})\right)$ shall be minimized, whereas d is a distance measurement like euclidean distance or Kullback-Leibler divergence and $\hat{\boldsymbol{x}}$ is the perturbed input. It is possible to sample $\boldsymbol{x}$ from both $\mathcal{D}_L$ and $\mathcal{D}_{UL}$.

An empirical investigation Oliver et al. (2018) has shown, that many consistency regularization methods, like mean teacher (Tarvainen & Valpola (2017)), Π-model (Sajjadi et al. (2016), Laine & Aila (2017)), and virtual adversarial training (VAT) Miyato et al. (2018) are quite hard to compare, since the results may rely on many parameters (network, task, etc.). Nevertheless VAT is chosen in this work, since it achieves convincing results on many tasks. VAT is a training method, which is greatly inspired by adversarial training (Goodfellow et al. (2015)). The perturbation $\boldsymbol{r}_{adv}$ of the input $\boldsymbol{x}$ can be computed as

$$\boldsymbol{r} \sim \mathcal{N}\left(0, \frac{\xi}{\sqrt{dim(\boldsymbol{x})}}\boldsymbol{I}\right) \tag{1}$$

$$\boldsymbol{r}_{adv} = \epsilon \frac{\nabla_{\boldsymbol{r}} d(f(\boldsymbol{x},\boldsymbol{\theta}), f(\boldsymbol{x}+\boldsymbol{r},\boldsymbol{\theta}))}{||\nabla_{\boldsymbol{r}} d(f(\boldsymbol{x},\boldsymbol{\theta}), f(\boldsymbol{x}+\boldsymbol{r},\boldsymbol{\theta}))||} \tag{2}$$

,where ξ and ϵ are hyperparameters, which have to be tuned for each task. After the perturbation was added to $\boldsymbol{x}$ consistency regularization is applied. The distance between the clean (not perturbed) prediction and perturbed prediction $d(f(\boldsymbol{x},\boldsymbol{\theta}), f(\boldsymbol{x}+\boldsymbol{r}_{adv},\boldsymbol{\theta}))$ shall be minimized. In order to reduce the distance the gradients are just backpropagated through $f(\boldsymbol{x}+\boldsymbol{r}_{adv})$. Combining VAT with entropy minimization Grandvalet & Bengio (2005) it is possible to further increase the performance Miyato et al. (2018). For entropy minimization an additional loss term is computed as:

$$-\sum f(\boldsymbol{x};\boldsymbol{\theta})\log[f(\boldsymbol{x};\boldsymbol{\theta})] \tag{3}$$

and added to the overall loss. This way the network is forced to make more confident predictions regardless of the input.

2.3 Generative Models

Generative models are commonly used for increasing the accuracy or robustness of models in a semi- or unsupervised manner (Kingma et al. (2014), Zhao et al. (2016), Springenberg (2015), Odena (2016), Radford et al. (2016)).

A popular approach is the use of generative adversarial neural networks (GANs), introduced by Goodfellow et al. (2014). The goal of a GAN is to train a generator network G, wich produces realistic samples by transforming a noise vector $\boldsymbol{z}$ as $\boldsymbol{x}_{fake} = G(\boldsymbol{z},\boldsymbol{\theta})$, and a discriminator network D, which has to distinguish between real samples $\boldsymbol{x}_{real} \sim p_{Data}$ and fake samples $\boldsymbol{x}_{fake} \sim G$.

In this paper the training method defined in Salimans et al. (2016) is used. Using this approach the output of D consists of $K+1$ categories, whereas K is the number of categories the classifier shall be actually trained on. One additional extra category is added for samples generated by D. Since the output of D is over-parameterized the logit output l_{K+1}, which represents the fake category, is permanently fixed to 0 after training. The loss function consists of two parts $L_{supervised}$ and $L_{unsupervised}$, which can be computed as:

$$L_{supervised} = -\mathbb{E}_{\boldsymbol{x},y\sim p_{data}} \log[p(y|\boldsymbol{x}, y < K+1)] \tag{4}$$

$$L_{unsupervised} = -\{\mathbb{E}_{\boldsymbol{x}\sim p_{data}} \log[1 - p(y = K+1|\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim G} \log[p(y = K+1|\boldsymbol{x})]\}. \tag{5}$$

$L_{supervised}$ represents the standard classification loss, i.e. negative log probability. $L_{unsupervised}$ itself again consists of two parts, the first part forces the network to output a low probability of fake category for inputs $\boldsymbol{x} \sim p_{data}$ and corresponding a high probability for inputs $\boldsymbol{x} \sim G$. Since the the category y is not used in $L_{unsupervised}$, the input $\boldsymbol{x}$ can be sampled from both $\mathcal{D}_L$ and $\mathcal{D}_{UL}$. In order to further improve the performance feature matching is used, as described in Salimans et al. (2016).

3 Experiments

Three different experiments are conducted in this paper using the MNIST (LeCun et al. (1998)) and UrbanSound8k Salamon et al. (2014) dataset. The UrbanSound8k dataset consists of 8732 sound clips with a maximum duration of $4\,$s. Each sound clip represents a different urban noise class like drilling, engine, jackhammer, etc. Before using the sound files for training a neural network, they are prepared in a similar manner to Salamon & Bello (2017), in essence each sound clip is transferred to a log-scaled mel-spectrogram with 128 components covering the frequency range between 0-22050 Hz. The window size is chosen to be $23\,$ms and hop size of the same duration. Sound snippets with shorter duration as $4\,$s are repeated and concatenated until a duration of $4\,$s is reached. The preprocessing is done using librosa (McFee et al. (2015)). For training and evaluation purposes a random snippet with a length of $3\,$s is selected, resulting in an input size of 128×128.

In the first experiment no external unlabeled data is used. Instead, the amount of labeled data in each category is varied and the three methods are compared to each other. In the second experiment the amount of labeled and unlabeled data is varied, in order to explore how unlabeled data can compensate labeled data. The last experiment considers class distribution mismatch while the amount of labeled and unlabeled data is fixed. In the second and third experiment only two methods are compared, since only generative models and consistency regularization allow the use of external unlabeled data.

All methods are compared to a standard model. When using the MNIST dataset the standard model consists of three convolutional layers, followed by two fully-connected layers. For the UrbanSound8k dataset the standard model consists of four convolutional layers, followed by three fully-connected layers. ReLU nonlinearities were used in all hidden layers. The training was done by using the Adam optimizer (Kingma & Ba (2014)). Furthermore batch normalization (Offe & Szegedy (2015)), dropout (Srivastava et al. (2014)), and max-pooling was used between convolutional layers. For further increasing the generalization capability L_2 regularization (Ng (2004)) is used. The models, representing the three different approaches, have the same computational power as the standard model, in essence three/ four convolutional layers and two/ three fully connected layers. The number of hidden dimensions and other per layer hyperparameters (e.g. stride, padding) is kept equal to the corresponding standard models. The hyperparameters were tuned manually on the training dataset by performing gridsearch and picking the most promising results. Whereas the L_2 and batchnorm coefficients, as well as dropout rate are shared across all models for each dataset. The test accuracy was calculated in all experiments with a separate test dataset, which contains 500 samples per category for the MNIST dataset and, respectively, 200 samples per category for the UrbanSound8k dataset. Train and test set have no overlap. All experiments were conducted using the PyTorch framework (Paszke et al. (2017)).

3.1 Varying amount of labeled Data and no unlabeled Data

In this experiment the amount of labeled data is varied. Furthermore there is not used any unlabeled external data. For each amount of labeled data and training approach (i.e. baseline, VAT Miyato et al. (2018), GAN Salimans et al. (2016), and siamese neural network Koch et al. (2015)) the training procedure was repeated eight times. Afterwards the mean accuracies and standard deviations have been calculated.

Figure 1 shows the results obtained in this experiment for the MNIST dataset. The amount of labeled data per category was varied on a logarithmic scale in the range between $[0, 200]$ with 31 steps. Using 200 labeled samples per category the baseline network is able to reach about $95\,\%$ accuracy. With just one labeled sample per class the baseline networks reaches already around $35\,\%$, which is a already good compared to $10\,\%$, when random guessing. Generally all three methods are consistent with the literature, such that they are superior over baseline in the low data regime (1-10 samples per category). Using a siamese neural network the accuracy can be significantly improved in the low data regime. With just one labeled sample the siamese architecture already reaches around $45\,\%$. When using a dataset with more categories, like Omniglot, the advantage of using siamese networks should be even higher in the low data regime. The performance of this approach becomes worse compared to the baseline model, when using more than 10 labeled examples per class. VAT has a higher benefit compared to GAN for up to 20 labeled samples per category. For higher numbers of labeled samples both methods show only little (0-2 %) improvement over the baseline results.

Similar results are obtained on the UrbanSound8k dataset (figure 2). As for the experiment on the MNIST dataset the amount of labeled data was varied on a logarithmic scale in the range between $[0, 200]$, but with 6 steps instead of 31, since the computational effort was much higher. The siamese network yields a large improvement when there is only one labeled sample, but fast returns worse results than the baseline network. On contrast the usage of VAT or GAN comes with a benefit in terms of accuracy for higher amounts of labeled data. Nevertheless these both methods are either not able to further improve the accuracy for high amounts of labeled data (more than 100). Furthermore the accuracy even declines compared to baseline for 200 labeled samples. The observation, that adversarial training can decrease accuracy, is inline with literature (Schmidt et al. (2018), Su et al. (2018)), where it was shown that in high data regimes there may be a trade-off between accuracy and robustness. Whereas in some cases adversarial training can improve accuracy in the low data regime.

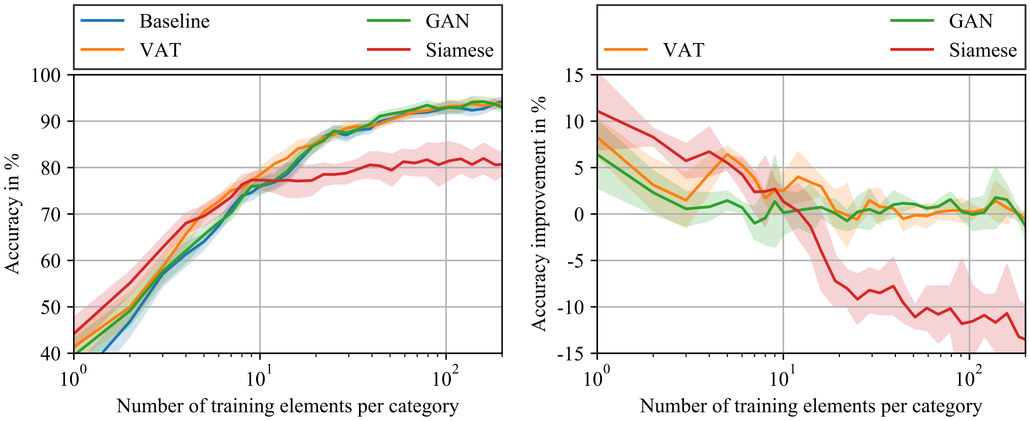

Figure 1: Comparison of different methods with varying amount of labeled data on MNIST. Left: Total accuracy achieved for each method. Right: Accuracy improvements over baseline.

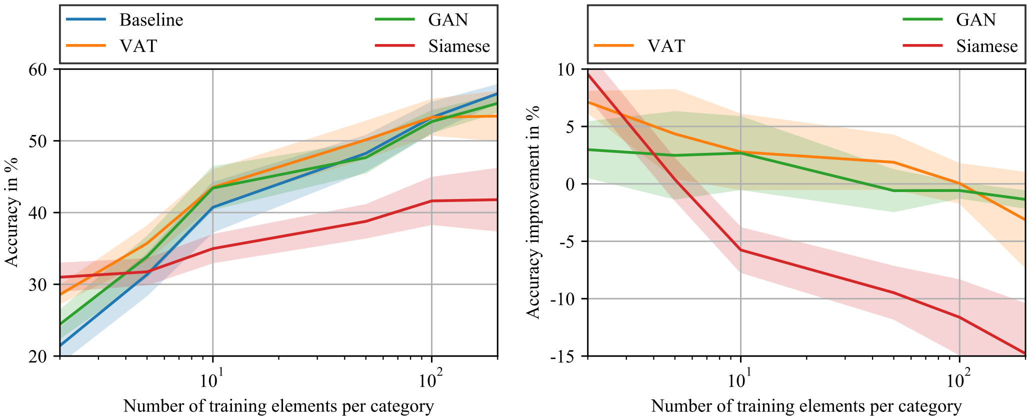

Figure 2: Comparison of different methods with varying amount of labeled data on UrbanSound8k. Left: Total accuracy achieved for each method. Right: Accuracy improvements over baseline.

3.2 Varying amount of labeled Data and unlabeled Data

In this experiment the amount of labeled and unlabeled data was varied. Like in the previous experiment the scale, on which the amount of data is varied, is chosen to be $[0, 200]$ with 31 steps for the MNIST dataset and 6 steps for the UrbanSound8k dataset. Since only the generative models and consistency regularization allow the use of unlabeled data, siamese neural networks have not been investigated in this experiment. Each of the two different approaches has been trained eight times for every point (number of labeled examples and unlabeled examples per category), afterwards the run with the highest result was picked as the final result. Baseline results have been computed in a similar way. For each amount of labeled data eight networks have been trained and the best result has been picked. Afterwards the difference between VAT/ GAN and baseline has been calculated.

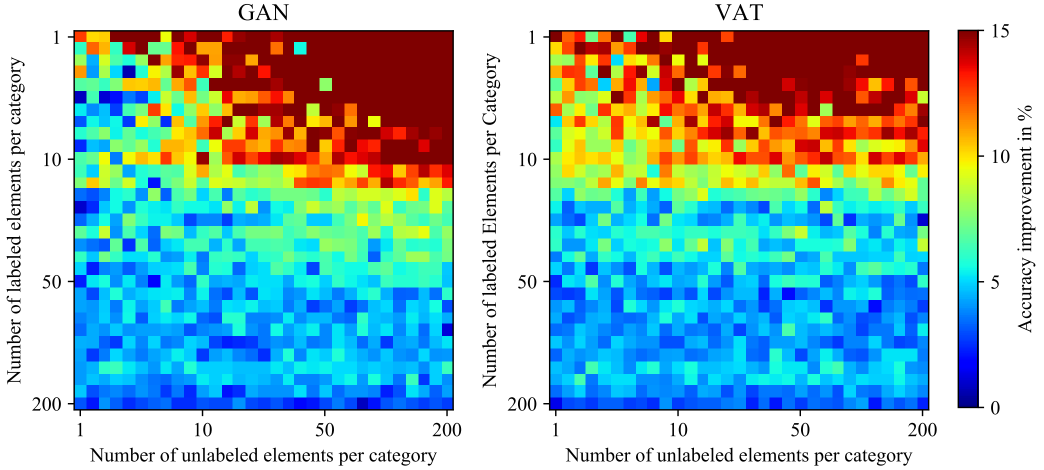

Figure 3: Comparison of VAT and GAN training with varying amount of labeled and additional unlabeled data to baseline method for MNIST dataset.

Figure 3 shows the results of this experiment for the MNIST dataset. On the left side the results achieved with the GAN method and on the right side the results achieved with VAT are visualized. Both methods show a significant increase in terms of accuracy when the amount of labeled data is low and corresponding the amount of unlabeled data is high. When the amount of labeled data increases the amount of necessary unlabeled data also increases in order to achieve the same accuracy improvements. VAT achieves better results with less unlabeled data compared to GAN, when there is little labeled data (∼ 2-10 examples per category). On contrast GANs achieve better results when there is a moderate amount of labeled examples (∼ 10-50 examples per category) and also many unlabeled examples. When the amount labeled examples is high both methods behave approximately equal.

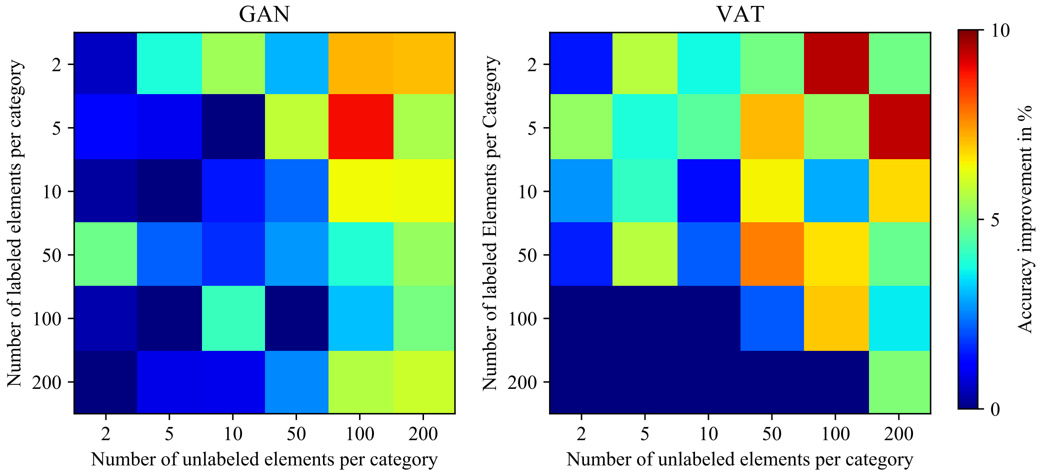

Figure 4: Comparison of VAT and GAN training with varying amount of labeled and additional unlabeled data to baseline method for UrbanSound8k dataset.

The results for the UrbanSound8k dataset can be seen in Figure 4. Overall similar results as for the MNIST dataset are achieved, in terms of having high benefits, when the amount of labeled data is low and concurrently the amounts of unlabeled data is high. Nevertheless the total improvement is lower and for high amounts of labeled data, more unlabeled data is necessary in order to get an improvement at all. For the VAT the amount of unlabeled data need to have similar magnitudes as the amount of labeled data in order to get an improvement at all. Further the same observation as before can be made, that VAT achieves better results with less unlabeled data, when there is little labeled data

3.3 Class Distribution Missmatch

In this experiment the possibility of adding additional unlabeled examples, which do not correspond to the target labels (mismatched samples), is investigated. This experiment was done for VAT in Oliver et al. (2018). In this work the investigation is extended in such a way that the results for VAT are compared to GAN. Furthermore not only the extend of mismatch, but also the influence of the amount of additional unlabeled examples is investigated. Both datasets (MNIST and UrbanSound8k) consist of 10 categories with label values $[0, 9]$ and the aim is to train a neural network, which is able to classify inputs corresponding to categories $[0, 6]$, hence the network has six outputs. Mismatched examples belong to categories $[7, 9]$. The number of labeled examples per category is fixed to be five. Having five labeled samples it can be seen in figure 3 and 4, that the accuracy improvement shows a strong dependency on the amount of unlabeled samples. The total number of unlabeled examples is varied between $\{30, 120, 600\}$. Furthermore the mismatch for each number of unlabeled examples is varied between 0-100 % using a 10 % increment, e.g. when the amount of unlabeled examples is set to be 120 and the mismatch is 70 % the unlabeled examples consist of 84 examples belonging to categories $[0, 6]$ and 36 examples belonging to categories $[7, 9]$. The distribution across the categories in the six matched and four remaining mismatched classes is kept approximately equal, with a maximum difference of ± 1. For each amount of mismatch and method eight neural networks have been trained. Afterwards their average accuracies and standard deviations have been calculated. For baseline results also eight neural networks have been trained and their average accuracy and standard deviation computed. Since the number of classes is reduced to 6 the accuracy, when compared to the previous experiments, is higher with the same amount of labeled data.

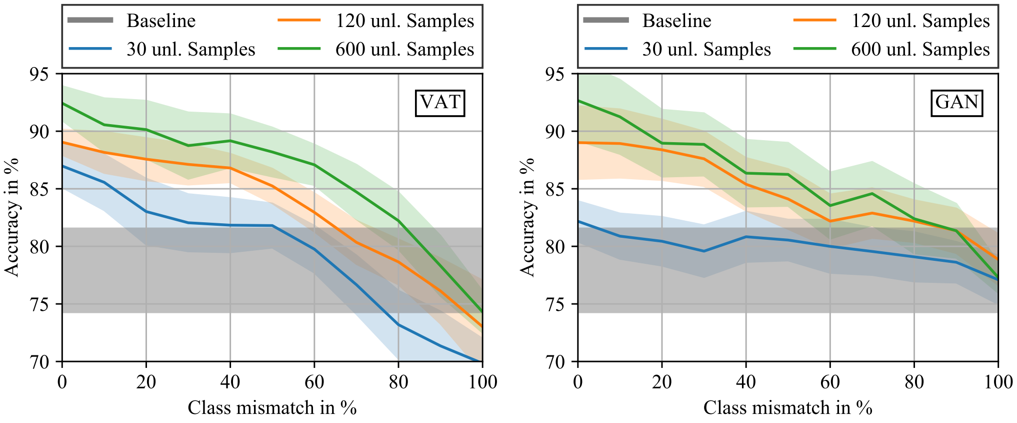

Figure 5: Comparison of different methods with varying amount of mismatch and unlabeled data for MNIST dataset. All networks have been trained with five labeled samples.

Figure 5 shows the results of this experiment for the MNIST dataset. Overall the accuracy decreases for both methods when the class mismatch increases, which is in line with literature (Oliver et al. (2018)). As in the experiments before, the GAN method shows little to no accuracy improvement, when the additional amount of unlabeled data is low (30 unlabeled samples). For 120 and respectively 600 additional unlabeled elements both methods show an approximate equal maximal accuracy improvement, when there is no class mismatch. When the class mismatch is very high

(80-100 %) using VAT results in worse performance than baseline results. Using GANs the performance is in worst case at the same level as baseline performance. GAN shows a linear correlation between accuracy and class mismatch. On contrast VAT shows a parabolic trend. Overall increasing the amount of unlabeled data seems to increase the robustness towards class mismatch. All in all both methods show an accuracy improvement even for high amounts ($> 50\,\%$) of class mismatch. Whereas VAT performs better, when the amount of mismatch is low.

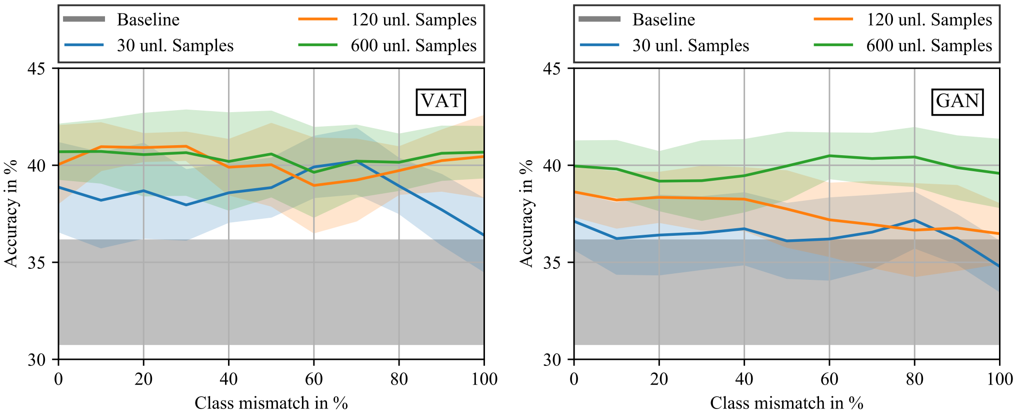

Figure 6: Comparison of different methods with varying amount of mismatch and unlabeled data for UrbanSound8k dataset. All networks have been trained with five labeled samples.

Figure 6 shows the results obtained with the UrbanSound8k dataset. Overall there seems to be no, or only little correlation between class mismatch and accuracy. Only for the GAN, when using 30 or 120 unlabeled samples, a small correlation can be observed. This is a surprising observation, since in the previous experiment and in Oliver et al. (2018) a decrease in terms of accuracy is reported for increasing class mismatch. In essence it can be stated, that adding samples, which do not necessarily belong to the target classes, can improve the overall accuracy. This is especially interesting for training classifiers on hard to obtain or rare samples (rare disease, etc.). Nevertheless it has to be checked whether adding this samples hurts the performance or not. Furthermore the correlation between more unlabeled data and accuracy can be observed, as in the previous experiments.

4 Conclusion

In this paper three methods for dealing with little data have been compared to each other. When the amount of labeled data is very little and no unlabeled data is available, siamese neural networks offer the best alternative in order to achieve good results in terms of accuracy. Furthermore when there is additional unlabeled data available using GANs or VAT offer a good option. VAT outperforms GAN when the amount of data is low. On contrast GANs should be preferred for moderate or high amounts of data. Nevertheless both methods must be tested for any individual use case, since the behavior of these methods may change for different datasets.

Surprising results have been obtained on the class mismatch experiment. It was observed that adding samples, which do not belong to the target classes, not necessarily reduce the accuracy. Whether adding such samples improves or reduce the accuracy, may heavily depend on how closely these samples/ classes are related to the target samples/ classes. An interesting questions remains whether datasets which perform good in transfer learning tasks (e.g. transferring from ImageNet to CIFAR-10) also may be suitable for such semi-supervised learning tasks.

Furthermore any combinations of three examined methods can bear interesting results, e.g.VAT could be applied to the discriminator in the GAN framework. Also a combination of GAN and siamese neural networks could be useful, in this case the siamese neural network would have two outputs, one for the source and one for the similarity.

References

Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning Workshop*, volume 2 of *UTLW'11*, pp. 17–37, 2011.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. 2014.

Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Ruslan R Salakhutdinov. Good semi-supervised learning that requires a bad gan. In *Advances in Neural Information Processing Systems*, pp. 6510–6520. 2017.

J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Ian Goodfellow, Jean Pouget Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. 2014.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*. 2015.

Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems 17*, pp. 529–536. 2005.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*. 2014.

Diederik Kingma, Danilo Rezende, Shakir Mohamed, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pp. 3581–3589. 2014.

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations*. 2017.

Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. One shot learning of simple visual concepts. In *CogSci*. 2011.

Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. pp. 2278–2324, 1998.

Brian McFee, Colin Raffel, Dawen Liang, Daniel P.W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and Music Signal Analysis in Python. In *Proceedings of the 14th Python in Science Conference*, pp. 18 – 25, 2015.

Takeru Miyato, Shin ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

Andrew Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pp. 78–, 2004.

Augustus Odena. Semi-supervised learning with generative adversarial networks, 2016. arxiv preprint arXiv:1410.5093.

Sergey Offe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*. 2015.

Avital Oliver, Augustus Odena, Colin Raffel, Ekin D. Cubuk, and Ian J. Goodfellow. Realistic evaluation of semi-supervised learning algorithms. In *International Conference on Learning Representations Workshop*. 2018.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations*. 2016.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems 29*, pp. 1163–1171. 2016.

J. Salamon, C. Jacoby, and J. P. Bello. A dataset and taxonomy for urban sound research. In *22nd ACM International Conference on Multimedia (ACM-MM'14)*, pp. 1041–1044, Orlando, FL, USA, 2014.

Justin Salamon and Juan Pablo Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, 24:279–283, 2017.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pp. 2234–2242. 2016.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. *CoRR*, abs/1804.11285, 2018.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 4077–4087. 2017.

Jost Tobias Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. In *International Conference on Learning Representations*. 2015.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. Is robustness the cost of accuracy? - A comprehensive study on the robustness of 18 deep image classification models. In *Computer Vision - ECCV 2018*, pp. 644–661, 2018.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, pp. 1195–1204. 2017.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pp. 3630–3638, 2016.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, pp. 3320–3328. 2014.

Junbo Zhao, Michael Mathieu, Ross Goroshin, and Yann Lecun. Stacked what-where auto-encoders. In *International Conference on Learning Representations Workshop*. 2016.